**Examiner's Amendment and Statement of Reasons for Allowance**

1. This action is responsive to Applicant's amendment filed on May 19, 2009.

## Information Disclosure Statement

2. The contents of information disclosure statement filed 03/18/2009 which are lined through fail to comply with the provisions of 37 CFR 1.98 because these contents cannot be listed in a printing patent. These contents which are the internal office actions replying by patent Examiners should not be listed in a printing patent.

***In accordance to 37 CFR 1.98 (a) (2) (ii) or (iv), the considered portions will be listed:***

> (ii) Each publication or that portion which caused it to be listed, other than U.S. patents and U.S. patent application publications unless required by the Office;
>
> (iv) All other information or that portion which caused it to be listed.

Thus, every content submitted under 37 CFR 1.98 when being initialed will cause to be listed in a printing patent. Therefore, pursuant to 37 CFR 1.98, Applicants should submit the contents as U.S. patent application publications and/or pending unpublished U.S. applications. An office action cannot be the portion which is caused to be listed in accordance to 37 CFR 1.98 (a) (2) (ii) or (iv).

As per request for considering the portions/contents that are the internal Examiner office actions, these portions/contents are considered by the Examiner, but they are lined through because of ***37 CFR 1.98 (a) (2) (ii) or (iv).***

## Examiner's Amendment

3. An examiner's amendment to the record appears below. Should the changes and/or additions be unacceptable to applicant, an amendment may be filed as provided by 37 CFR 1.312. To ensure consideration of such an amendment, it MUST be submitted no later than the payment of the issue fee.

Authorization for this examiner's amendment was given in a telephone interview with Mr. Jens Jenkins, Registration Number 44,803, on August 14, 2009 for obviating any potential 101 issues and put the claims in condition for allowance.

The application has been amended as follows:

1. (Currently amended) A preference execution system comprising:

a memory;

a data store component for storing schematized data and end-user specified preferences, wherein queries are evaluated and stored as data in the data store component, and constructed upon demand;

a compiler to compile information agent applications including end-user specified preferences and store the compiled information agent applications in the data store;

an execution engine to retrieve preferences stored in the data store upon the occurrence of one or more events and to utilize the preferences and at least one stored procedure to query tables within the data store and produce a results table, wherein the results table stores preferences whose conditions have been satisfied such that specified actions are triggered based on the stored preferences; and

a context analyzer that stores and analyzes information regarding variables and parameters of a user that influence notification decision-making, the parameters comprise a user's typical locations and attentional focus, devices users tend to have access to in different locations, a user's preference as to being disturbed by notifications of different types in different settings, default parameters as to how the user wishes to be notified in different situations, [[or]] and cost of disruption associated with being notified by different modes in different settings, the parameters are stored as a user profile that can be edited by the user or users can specify in real-time their state; and

a constant accessor that provides navigation to data across different domains to enable a user to write cross-domain preferences.

Claims 14-17 (Canceled).

Claim 18 (Currently Amended) A method for employing preferences comprising:

specifying user preferences regarding an information agent application based on a developer schema;

storing the preferences and schematized data in one or more tables in a data store;

querying the tables in the data store upon occurrence of an event and retrieving preferences stored in the data store;

producing a results table, wherein the results table stores preferences whose conditions have been satisfied such that specified actions are triggered;

executing actions based on the results table;

utilizing a context analyzer that stores and analyzes information regarding variables and parameters of a user that influence notification decision-making, the parameters comprise a user's typical locations and attentional focus, activities per time of day and day of week, devices users tend to have access to in different locations, a user's preference as to being disturbed by notifications of different types in different settings, [[or]] and cost of disruption associated with being notified by different modes in different settings; and

storing the parameters as a user profile that can be edited by the user or allowing users to specify in real-time their state;

utilizing a one-at-a-time declarative programming model, wherein user preferences are specified using one or more On-event-If-Then statements and Boolean operators to specify conditions and actions, schema; and

providing a constant accessor that allows preferences and conditions to be written that are capable of navigating and retrieving information from various domains.

Claim 26 (Canceled).

-- The End --

## Examiner's Statement of Reason(s) for Allowance

5. Claims 1-6, 8-13, 18, 23-25 are allowed.

6. The following is an examiner's statement of reasons for allowance:
The prior arts of record: **Black** et al., teaches a method and apparatus for accessing network device data through user profiles. User profiles may be created by network administrators, and the corresponding user profile data may be stored In a central network management system (NMS) database. When a user requests data from a particular network device, the NMS utilizes the user profile data in the central database to access the network device and retrieve the required network device data. Since the user profile data is stored in a central database, the user may log into the NMS from any location and connect to any network device in the network. **Abbott** et al., teaches a characterization module receives an invocation request to provide an attribute value that was generated by a requesting attribute consumer. The received invocation request identifies the attribute whose value is to be provided. In response to receiving the invocation request, the characterization module provides a value for the identified attribute to the requesting attribute consumer. **Josephson** teaches aspects of the invention provide for interfacing one or more users or groups of users with one or more machines or groups of machines in a manner adaptable to conventional and non-conventional control and data entry capabilities, and that can be used alone and in conjunction with other interfacing elements, approaches and operabilities. Complete interfaces, command sets, commands, feedback, and new environments and operations are also enabled in a modifiable, extensible, broadly applicable and portable manner that is capable of exploiting flexibility, suggestibility and other aspects of human

expression. **Seshadri** '916 et al., teaches a system and methodology to facilitate timely delivery of notifications and performance of automated actions for users in the framework of an ad-hoc rules processing environment. An information agent operates as a distributed application to perform automated actions for users in accordance with user preferences and context. Various rules can be executed to delay, inhibit, and/or enable an action such as sending notifications according to the preferences and current user states as determined from the context. The rules can include data centric matching logic, set-oriented execution rules, rule ordering/interaction hierarchies, and ad-hoc Boolean combinations.
**Seshadri** '202 et al., teaches a system and methodology to facilitate development, debug, and deployment of a notification platform application. This includes providing various editor and display tools that interact with diverse components of the application in the context of a dynamic framework and development environment adapted to the nuances of the components. Visual displays and interfaces are provided that depict the various components of an application along with the relationships between the components. Developers are then directed though the components of the application in a visual manner, wherein various editors can be invoked to design a portion of the application relating to the selected component. Other aspects include providing a debug environment for troubleshooting a notification application along with deployment options to install and execute the application. **Ku** et al., teaches a system, method and user interface for visually browsing and editing one more or CORBA Interface Repositories ("IR") for program objects. A user selects an Interface Repository and an object within the Interface Repository. The IR Browser/Editor presents a three-pane view graphically depicting a containment tree, an object's interface definition language,

and an inheritance diagram of the selected object. The user may review the full interface definition language of the object, save the IDL or cut-and-paste it to another program such as a code editor for compiling. The user may also select a parent or child object within any of the three panes of the display, which causes the inheritance diagram, containment tree, and IDL panes to be updated to show the parent and child interfaces of the selected object, and the containment tree and IDL for the selected object, accordingly. **Thuraisingham**, teaches a Apparatus for designing a multilevel secure database management system based on a multilevel logic programming system. The apparatus includes a multilevel knowledge base which has a multilevel database in which data are classified at different security levels. The multilevel knowledge base also includes schema, which describe the data in the database, and rules, which are used to deduce new data. Also included are integrity constraints, which are constraints enforced on the data, and security constraints, which are rules that assign security levels to the data. The system further includes users cleared to the different security levels for querying the multilevel database, and a multilevel logic programming system is provided for accessing the multilevel knowledge base for processing queries and for processing the integrity and security constraints. **Saxe** et al., teaches a system and method for increasing the speed of operation of a theorem prover relating to program verification using adaptive pattern matching technique is disclosed. Source code in a specific programming language is converted to one or more formulae, each representing a specific reformulation of the source code that facilitates program verification. Each formula derived from the source code is converted into an E-graph which is a particular type of a directed acyclic graph having leaf nodes and interior nodes. Some of the nodes of an E-graph may be related to other nodes through equivalence relationships. Equivalence relationships between a group of nodes is stored in a data structure called an equivalence class. A collection of rules defining the grammar of the programming language is stored in an axiom database. Rules and conjectures can dynamically be added to the axiom database. Each rule or conjecture to be tested is converted into a pattern. **Gram**, teaches a method for allowing a user to customize an interface for a computer program are provided. The methods allow a user of the computer program flexibility in organizing commands into a menu structure. The methods also allow a user of a computer program to assign a keystroke sequence to a command. The user can then invoke the command by entering the keystroke sequence. A command assigned to a keystroke sequence does not need to be associated with a menu. The methods allow the user of a computer program to predefine parameters for commands. The predefinition allows the user to invoke a command without reentering the parameters. In a preferred embodiments, the methods use a command array that contains an entry for each command. The methods use an array associated with each menu. The arrays contain unique identifiers of the commands that are associated with the menu. These unique identifiers are used to reference entries into the command array. The entries in the command array include a reference to the program code that implements the commands. **Sutter**, teaches an independent distributed database system comprising a plurality of sites wherein all users at all sites work off-line with local data. All application transactions are against the local database only, and every site stores "all and only" the data it needs. On-line transactions occur only in the background, including a periodical "synch" between sites that transmits any changes to data of interest to that site. If the background operations are interrupted or the network is temporarily unavailable, the user does not see new changes made at other sites until the data link is available again, but is otherwise unaffected. It is a feature that no site acts as a "server" for any other site. Some sites may store more data or have more users than others, but all sites are logically peers. **Alumbaugh** et al., teaches a system, including software components, that efficiently and dynamically analyzes changes to data sources, including application programs, within an integration environment and simultaneously re-codes dynamic adapters between the data sources is disclosed. The system also monitors at least two of said data sources to detect similarities within the data structures of said data sources and generates new dynamic adapters to integrate said at least two of said data sources. The system also provides real time error validation of dynamic adapters as well as performance optimization of newly created dynamic adapters that have been generated under changing environmental conditions. **Yamanoue** et al., teaches a data supply controlling device, a data supplying method, a storage medium storing a data supplying program, and a data supplying system, in which a system is used which searches for information (of books, for instance) using a data supplying device and provides search results to the user (user terminal) via the data supply controlling device in such a manner to provide information suitable for user's preference and interest by referring to user data and also to achieve improvement in both user privacy protection and communications efficiency. **Delo**, teaches methods for Optimizing the Installation of a Software Product onto a Target Computer System Optimizations for the process of installing a software product onto a target computer system. A relational installation database for storing data elements in the form of strings, objects, etc. is aliased with integer identifiers corresponding to each data element. The integer identifiers are obtained from an index that sequentially stores a copy of each unique occurrences of a data element. Populating an installation database with only integers reduces persistent size and provides uniformity to the data fields underlying the database tables, and provides a significant improvement in database performance. The uniform data fields may be expanded and contracted to add temporary rows and columns directly to a database table. In this way, temporary data elements may be stored directly in a database table without the need for creating a view of the database table. Temporary data elements are lost when the database table is no longer referenced, and temporary data is ignored when persisting the database. Aliased installation databases are easily manageable and provide greater flexibility to software developers in the creation and shipping of a software product. Database tables may be created in a modular fashion and may be efficiently merged together when the software product is complete. **Omoigui**, teaches an invention is directed to an integrated implementation framework and resulting medium for knowledge retrieval, management, delivery and presentation. The system includes a first server component that is responsible for adding and maintaining domain-specific semantic information and a second server component that hosts semantic and other knowledge for use by the first server component that work together to provide context and time-sensitive semantic information retrieval services to clients operating a presentation platform via a communication medium. Within the system, all objects or events in a given hierarchy are active Agents semantically related to each other and representing queries (comprised of underlying action code) that return data objects for presentation to the client according to a predetermined and customizable theme or "Skin." This system provides various means for the client to customize and "blend" Agents and the underlying related queries to optimize the presentation of the resulting information. **Keil** et al., teaches a system to determine preference information of part worth values associated with a consumer and a product includes determination of a plurality of attributes of the product, each of the plurality of attributes associated with a plurality of attribute levels, determination of a plurality of piles of attributes based on a first indication of the consumer, each of the plurality of piles comprising one or more of the plurality of attributes, determination of a ranked order of a plurality of attributes of one of the plurality of piles, determination of a relative importance of one or more of the plurality of attributes of the one of the plurality of piles based on a second indication of the consumer, determination of a scale value of one or more attribute levels of the one or more of the plurality of attributes of the one of the plurality of piles based on a third indication of the consumer, and determination of a part worth value associated with an attribute level of one of the plurality of attributes of the one of the plurality of piles based on a determined scale value of the attribute level and a determined relative importance of the one of the plurality of attributes. **Kagalwala** et al., teaches chema for a SQL (structured query language) database defines classes, properties, methods, and associations. A class models a set of objects that have similar properties and fulfill similar purposes. **Knutson** et al., teaches a system and method for allowing a user to segment and partition a database based upon attributes associated with the data in the database. Also, a system and method for generating a report for a user which allows the user to make decisions, without requiring the user to understand or interpret data itself. A database computer includes a database containing the data.

The data includes a collection of information about an enterprise of the user. A server computer is coupled to the database computer and executes a database management program. A client computer is coupled to the server and executes an application program. The application program allows a user to define predetermined data types, to define relationships between the data types, to define parameters for the report, to define a method of analysis for the report, and to create the report. The report summarizes the data in terms of the data types, the data relationships, and the method of analysis. **Knight** et al., teaches systems and methods disclose a system for personalizing computer functionality. End-users are provided with tools to easily write rich and complex preferences, for example, by using a plurality simple IF-THEN propositional logic. The preferences are then transformed into queries and executed efficiently on structured data. Preferences that are satisfied then execute actions such as providing notification or storing data in a particular folder. Furthermore, according to an aspect of the invention, data, logic, events, inter alia, are all schematized, thereby enabling sharing of data between application components and across applications. **Jammes** et al., teaches system and method for designing and operating an electronic store (1) permit a merchant to organize and advertise descriptions of product inventory over the Internet, (2) permit Web page information to be extracted on-demand from a product inventory database, and (3) permit Web pages to be automatically customized to fit shopping behaviors of individual consumers. A graphical store design user interface of a Web browser displays a hierarchical representation of products and, product groups of an electronic store. A user manipulates icons of the Web browser store design user interface to cause a Web server to modify relationships between products and product groups stored in a product information database. A store designer creates HTML template files, embeds database and customize references within the template files, and assigns template files to groups or products of the electronic store. **Bennett**, teaches a method of presenting catered information to a user. An item to be output to the user is identified and sub-items are retrieved from at least one storage. The system builds output from the sub-items and presents the output to the user. User interface activity is recorded in storage and used to modify the user interface. And **Bailey** et al., teaches "Event-condition-action" (ECA) rules are a technology from active databases and are a natural method for supporting such functionality. ECA rules can be used for activities such as automatically enforcing document constraints, maintaining repository statistics, and facilitating publish/subscribe applications. New arts made of record: US Patent No. 6,490,718 by **Watters**, teaches a method for processing electronic data is provided. The method includes creating a visual analog of a data format for the electronic data and converting the visual analog into data processing commands. Electronic data may then be processed with the data processing commands. US Patent No. 6,826,541 by **Johnston**, teaches methods, systems, and computer program products for facilitating user choices among complex alternatives utilize conjoint analysis to simplify choices to be made by the user. A selector tool presents a user with a first and second series of choices relating to attributes of products or services available to the user. A utilities calculation engine calculates the relative utility of each of the products or services to the user and presents output to the user, which indicates the relative utility of each of the products or services. The user can then select the product or service that has the highest utility value for the user based on the calculated relative utility values. And US 2005/0084082 by Horvitz et al., teaches an invention relates to utilizing identity and context-sensitive decision-making for handling communications, including, channel selection, routing, and rescheduling operations. The systems and methods provide a service that allows users to assess preferences regarding real-time call handling and performs dynamic decision-making about the best timing and channel for interpersonal communication. This service can be based on various cost-benefit analyses (e.g., basic and extended) that consider cost of interruption and preferences of contactors and contactees to guide communications, and/or on decision-making under uncertainty. Statistical models that are learned from data are joined with user preferences to generate expected costs of interruption for office activity and over time, based on a user's activities, locations, calendar information and preference assessments. However, none of them, taken alone or in combination, teaches the features in such a manner as recited in independent claims 1, and 18.

Any comments considered necessary by applicant must be submitted no later than the payment of the issue fee and, to avoid processing delays, should preferably accompany the issue fee. Such submissions should be clearly labeled "Comments on Statement of Reasons for Allowance."

5. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Chih-Ching Chow whose telephone number is 571-272-3693. The examiner can normally be reached on 8:00am - 4:30pm.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Wei Zhen can be reached on 571-272-3708. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

/Chih-Ching Chow/
Examiner, Art Unit 2191
08/14/2009
/Wei Y Zhen/
Supervisory Patent Examiner, Art Unit 2191